II. AMENDMENTS

Specification

Applicants previously amended the Specification, Continuation Data in their Response to Non-Final Office Action dated 1/25/05, mailed on July 25, 2005, and date-stamped by the Patent Office on July 28, 2005. Therefore, Applicants respectfully request the Examiner to refer to such document and update accordingly.

Claims

Please cancel rejected claims 1-3, 5-23, 26-38, 40-46 and 119-121. Please amend claims 24, 25, and 122 as follows:

1-3. (Cancel)

4. (Previously cancelled)

5-23. (Cancel)

24. (Currently amended) ~~The method of claim 1, further comprising:~~ A method for storing data in a memory in a computer system, the method comprising:

receiving uncompressed data;

determining a compression mode for the data, wherein the compression mode comprises one of lossless compression, lossy compression, or no compression;

selectively compressing the uncompressed data, wherein said compressing is selectively performed in response to the compression mode for the data;

storing the data in the memory;

creating a header after said determining the compression mode for the data, wherein the header includes compression mode information indicating the compression mode of the first data, wherein the compression mode information indicates a decompression procedure for decompression of the compressed first data; and

wherein said storing the data in the memory includes storing the header in the memory with the data.

25. (Original) The method of claim 24, further comprising:

receiving a request for the data;

accessing the data from the memory in response to the request;

analyzing the header to determine a compression mode for the data in response to receiving the request;

selectively decompressing the data, wherein said decompressing is selectively performed in response to the compression mode for the data; and

providing the data in response to the request.

26-38. (Cancel)

40-46. (Cancel)

47-57. (Previously cancelled)

58. (Original) A method for compressing data and storing the compressed data in a memory in a computer system, the method comprising:

receiving uncompressed first data;

compressing the uncompressed first data to produce compressed first data, wherein said compressed first data has a first size;

determining if the first size of the compressed first data is greater than an allocated memory block size of a first allocated memory block;

creating a header, wherein the header includes an overflow indicator indicating whether the first size of the compressed first data is greater than the allocated memory block size; and

storing the compressed first data and the header in the memory.

59. (Original) The method of claim 58, wherein said determining determines that the first size of the compressed first data is less than or equal to the allocated memory block size;

wherein the overflow indicator indicates that the first allocated memory block stores all of the compressed first data.

60. (Original) The method of claim 59, wherein said overflow indicator indicates that the last symbol of the compressed first data is stored in the first allocated memory block.

61. (Original) The method of claim 58, wherein said determining determines that the first size of the compressed first data is greater than the allocated memory block size;

wherein the overflow indicator indicates that the first allocated memory block does not store all of the compressed first data;

the method further comprising:

allocating a first overflow memory block;

storing overflow information in the header, wherein the overflow information includes an overflow address pointer which points to the first overflow memory block;

wherein said storing comprises:

storing a first portion of the compressed first data and the header in the first allocated memory block; and

storing an overflow portion of the compressed first data in the first overflow memory block.

62. (Original) The method of claim 61, wherein the first overflow memory block has a fixed size.

63. (Original) The method of claim 61, further comprising:

determining whether the overflow portion has a size greater than the first overflow memory block;

creating an overflow header, wherein the overflow header includes an overflow indicator indicating whether the overflow portion has a size greater than the first overflow memory block;

wherein said storing the overflow portion includes storing the overflow portion and the overflow header in the first overflow memory block.

64. (Original) The method of claim 63, further comprising:

wherein said determining determines that the overflow portion of the compressed first data has a size greater than the first overflow memory block;

wherein the overflow indicator in the overflow header indicates that the first overflow memory block does not store all of the overflow portion;

the method further comprising:

allocating a second overflow memory block in response to determining that the overflow portion of the compressed first data is greater than the first overflow memory block;

storing overflow information in the first overflow header, wherein the overflow information includes an overflow address pointer which points to the second overflow memory block;

wherein said storing comprises:

storing a first portion of the compressed first data and the header in the first allocated memory block;

storing a first overflow portion of the compressed first data in the first overflow memory block; and

storing a second overflow portion of the compressed first data in the second overflow memory block.

65. (Original) The method of claim 58, wherein said determining determines that the first size of the compressed first data is greater than the allocated memory block size;

wherein the overflow indicator indicates that the first allocated memory block does not store all of the compressed first data;

the method further comprising:

allocating a plurality of overflow memory blocks, including a first overflow memory block and a last overflow memory block;

storing overflow information in the header, wherein the overflow information includes an overflow address pointer which points to a first overflow memory block;

wherein said storing comprises:

storing a first portion of the compressed first data and the header in the first allocated memory block; and

for each of the overflow memory blocks except the last overflow memory block, storing, in the respective overflow memory block, an overflow portion of the compressed first data and a header pointing to a subsequent overflow memory block.

66. (Original) The method of claim 58, wherein said determining determines that the first size of the compressed first data is greater than the allocated memory block size;

wherein the overflow indicator indicates that the first allocated memory block does not store all of the compressed first data;

the method further comprising:

allocating one or more overflow memory blocks, wherein the first allocated memory block and the one or more overflow memory blocks are insufficient to store the compressed first data;

generating an interrupt to a driver in response to the first allocated memory block and the one or more overflow memory blocks being insufficient to store the compressed first data;

the driver allocating additional overflow memory blocks in response to the interrupt.

67. (Original) The method of claim 58, wherein said determining determines if the first size of the compressed first data and a maximum header size are greater than the allocated memory block size.

68. (Original) The method of claim 58, further comprising:

allocating the first allocated memory block in response to receiving the uncompressed first data, wherein the first allocated memory block is allocated according to a pre-determined compression ratio.

69. (Original) The method of claim 58, wherein the computer system includes an operating system, the method further comprising:

the operating system allocating the first allocated memory block in response to receiving the uncompressed first data.

70. (Original) A computer system including a memory controller having an embedded compression/decompression engine, the computer system comprising:

a CPU;

system memory which stores data used by said CPU for executing one or more applications;

a memory controller coupled to said system memory and said CPU, wherein said memory controller performs memory control functions for said system memory, wherein said memory controller includes said compression/decompression engine comprised in said memory controller for compressing and decompressing data transferred to or from said system memory;

wherein said memory controller is operable to:

receive uncompressed first data;

selectively compress the uncompressed first data to produce compressed first data according to a compression mode;

create a header, wherein the header includes compression mode information indicating the compression mode of the first data, wherein the compression mode information indicates a decompression procedure for decompression of the compressed first data; and store the compressed first data and the header in the memory.

71-94. (Previously cancelled)

95. (Original) A method for compressing data and storing the compressed data in a memory in a computer system, the method comprising:

allocating a memory block, wherein the memory block is allocated for uncompressed data;

receiving uncompressed first data;

receiving one or more destination addresses indicating a storage destination of the first data in the allocated memory block;

compressing the uncompressed first data to produce compressed first data;

storing the compressed first data in the allocated memory block at the one or more destination addresses.

96. (Original) The method of claim 95, wherein said storing does not perform address translation of the one or more destination addresses, wherein said storing provides reduced latency.

97. (Original) The method of claim 95, wherein the uncompressed first data has a first size, wherein the compressed first data has a second smaller size;

wherein said storing does not perform address translation of the one or more destination addresses, wherein said storing does not perform memory minimization.

98. (Original) The method of claim 95, wherein the computer system includes an operating system, wherein the operating system allocates the memory block for uncompressed data;

wherein the operating system does not account for the compression operation.

99. (Original) The method of claim 95, wherein the computer system includes an operating system, wherein the operating system allocates the memory block for uncompressed data;

wherein the operating system is unaware of the compression operation.

100. (Original) The method of claim 95, wherein the compressed first data occupies a first portion of the allocated memory block, the method further comprising:

allocating a portion of the allocated memory block as overflow storage.

101. (Original) The method of claim 100,

wherein the uncompressed first data comprises a plurality of blocks each having an original size, wherein one or more of the blocks compress to a larger size than the original size;

wherein said storing the compressed first data includes storing overflow data in the portion of the allocated memory block allocated as overflow storage.

102. (Original) The method of claim 95, wherein the uncompressed first data comprises application data generated by a CPU in the computer system.

103. (Original) The method of claim 95, wherein the memory comprises a system memory which stores application data generated by a CPU in the computer system.

104. (Original) The method of claim 95, further comprising

receiving a request for the first data;

decompressing the compressed first data to produce uncompressed first data;

providing the uncompressed first data in response to the request.

105. (Original) The method of claim 95, further comprising

receiving a request for the first data, wherein the request includes the one or more destination addresses in the allocated memory block where the compressed first data is stored;

accessing the compressed first data from the memory using the one or more destination addresses;

decompressing the compressed first data to produce uncompressed first data; and

providing the uncompressed first data in response to the request.

106. (Original) The method of claim 95, wherein the computer system includes a memory controller, wherein the memory controller performs said receiving uncompressed first data, said receiving one or more destination addresses, said compressing the uncompressed first data to produce compressed first data, and said storing the compressed first data.

107. (Original) A method for compressing data and storing the compressed data in a memory in a computer system, wherein the computer system includes an operating system, the method comprising:

the operating system allocating a memory block, wherein the operating system allocates the memory block for uncompressed data;

receiving uncompressed first data;

receiving one or more destination addresses indicating a storage destination of the first data in the allocated memory block;

compressing the uncompressed first data to produce compressed first data;

storing the compressed first data in the allocated memory block at the one or more destination addresses, wherein said storing does not perform address translation of the one or more destination addresses for reduced latency;

wherein the operating system does not account for the compression operation.

108. (Original) The method of claim 107, wherein the uncompressed first data has a first size, wherein the compressed first data has a second smaller size;

wherein said storing does not perform address translation of the one or more destination addresses, wherein said storing does not perform memory minimization.

109. (Original) A computer system utilizing compressed storage of data, the computer system comprising:

a CPU;

system memory which stores data used by said CPU for executing one or more applications, wherein the system memory also stores an operating system;

a memory controller coupled to said system memory and said CPU, wherein said memory controller performs memory control functions for said system memory, wherein said memory controller includes a compression/decompression engine comprised in said memory controller for compressing and decompressing data transferred to or from said system memory;

wherein memory blocks are allocated in the system memory for uncompressed data;

wherein the memory controller is operable to:

receive uncompressed first data;

receive one or more destination addresses indicating a storage destination of the first data in an allocated memory block;

compress the uncompressed first data to produce compressed first data; and

store the compressed first data in the allocated memory block at the one or more destination addresses.

110. (Original) The computer system of claim 109, wherein, in storing the compressed first data, the memory controller does not perform address translation of the one or more destination addresses, wherein the memory controller provides reduced latency.

111. (Original) The computer system of claim 109, wherein the uncompressed first data has a first size, wherein the compressed first data has a second smaller size;

wherein the memory controller does not perform address translation of the one or more destination addresses, wherein the memory controller does not perform memory minimization.

112. (Original) The computer system of claim 109, wherein the computer system includes an operating system, wherein the operating system allocates the memory block for uncompressed data;

wherein the operating system does not account for the compression operation.

113. (Original) The computer system of claim 109, wherein the computer system includes an operating system, wherein the operating system allocates the memory block for uncompressed data;

wherein the operating system is unaware of the compression operation.

114. (Original) The computer system of claim 109, wherein the compressed first data occupies a first portion of the allocated memory block;

wherein the memory controller is operable to allocate a portion of the allocated memory block as overflow storage.

115. (Original) The computer system of claim 114,

wherein the uncompressed first data comprises a plurality of blocks each having an original size, wherein one or more of the blocks compress to a larger size than the original size;

wherein the memory controller is operable to store overflow data in the portion of the allocated memory block allocated as overflow storage.

116. (Original) The computer system of claim 109, wherein the uncompressed first data comprises application data generated by the CPU.

117. (Original) The computer system of claim 109, wherein the memory controller is further operable to:

receive a request for the first data;

decompress the compressed first data to produce uncompressed first data; and

provide the uncompressed first data in response to the request.

118. (Original) The computer system of claim 109, wherein the memory controller is further operable to:

receive a request for the first data, wherein the request includes the one or more destination addresses in the allocated memory block where the compressed first data is stored;

access the compressed first data from the system memory using the one or more destination addresses;

decompress the compressed first data to produce uncompressed first data; and

provide the uncompressed first data in response to the request.

119-121. (Cancel)

122. (Currently amended) ~~The method of claim 119, wherein the uncompressed first data has a first size, wherein the compressed first data has a second smaller size;~~

~~the method further comprising:~~ A method for compressing data and storing the compressed data in a memory in a computer system, the method comprising:

allocating a memory block, wherein the memory block is allocated according to a pre-determined compression ratio;

receiving uncompressed first data having a first size;

receiving one or more destination addresses indicating a storage destination of the first data in the allocated memory block;

compressing the uncompressed first data to produce compressed first data having a second smaller size;

storing the compressed first data in the allocated memory block at the one or more destination addresses;

determining if the compressed first data fits within the allocated memory block; and

allocating an overflow memory block if the compressed first data does not fit within the allocated memory block.

III. CONCLUSION

In view of the amendments and remarks set forth herein, this application is believed to be in condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the Examiner is requested to telephone the undersigned.

Respectfully submitted,

Michael Adams

Michael P. Adams
Attorney for Applicants
Reg. No. 34,763
512.370.2858

Winstead Sechrest & Minick P.C.
P.O. Box 50784
Dallas, Texas 75201

CERTIFICATION UNDER 37 C.F.R. § 1.8

I hereby certify that this correspondence (along with any item referred to as being enclosed herewith) is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on December 12, 2005.

Michael Adams
Signature